LPC1100 系列微控制器

第十章 I^2C 接口

用户手册 Rev1.00

广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4

网址：http://www.zlgmcu.com

销售与服务网络

广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4 邮编：510630

电话：(020)38730972 38730976 38730916 38730917 38730977

传真：(020)38730925

网址：http://www.zlgmcu.com

广州专卖店

地址：广州市天河区新赛格电子城 203-204 室

电话：(020)87578634 87569917

传真：(020)87578842

南京周立功

地址：南京市珠江路 280 号珠江大厦 2006 室

电话：(025)83613221 83613271 83603500

传真：(025)83613271

北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座 1207-1208 室（中发电子市场斜对面）

电话：(010)62536178 62536179 82628073

传真：(010)82614433

重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦（赛格电子市场）1611 室

电话：(023)68796438 68796439

传真：(023)68796439

杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)28139611 28139612 28139613
28139615 28139616 28139618

传真：(0571)28139621

成都周立功

地址：成都市一环路南二段 1 号数码同人港 401 室（磨子桥立交西北角）

电话：(028) 85439836 85437446

传真：(028) 85437896

深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室

电话：(0755)83781788（5 线）

传真：(0755)83793285

武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室（华中电脑数码市场）

电话：(027)87168497 87168297 87168397

传真：(027)87163755

上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室

电话：(021)53083452 53083453 53083496

传真：(021)53083491

西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881296 83063000 87881295

传真：(029)87880865

目 录

第 10 章 I²C总线接口 3
10.1 本章导读 3
10.2 特性 3
10.3 应用 3
10.4 描述 3
10.4.1 I²C快速模式Plus 4
10.5 管脚描述 4
10.6 计时和功率控制 4
10.7 寄存器描述 4
10.7.1 I²C控制置位寄存器 6
10.7.2 I²C状态寄存器） 7
10.7.3 I²C数据寄存器 7
10.7.4 I²C从地址寄存器（I2ADR0 - 0x4000 000C） 7
10.7.5 I²C SCL高电平占空比寄存器和低电平占空比寄存器 8
10.7.6 I²C控制清零寄存器 8
10.7.7 I²C监控模式控制寄存器 9
10.7.8 I²C从地址寄存器 10
10.7.9 I²C数据缓冲寄存器 10
10.7.10 I²C屏蔽寄存器 10
10.8 I²C操作模式 11
10.8.1 主发送模式 11
10.8.2 主接收模式 12
10.8.3 从接收模式 12
10.8.4 从发送模式 13
10.9 功能描述 13
10.9.1 输入滤波器和输出级 13
10.9.2 地址寄存器 I2ADDR0~I2ADDR3 14
10.9.3 地址屏蔽寄存器，I2MASK0~I2MASK3 15
10.9.4 比较器 15
10.9.5 移位寄存器，I2DAT 15
10.9.6 仲裁及同步逻辑 15
10.9.7 串行时钟发生器 16
10.9.8 时序和控制 16
10.9.9 控制寄存器I2CONSET和I2CONCLR 16
10.9.10 状态解码器和状态寄存器 16
10.10 I²C操作模式详解 17
10.10.1 主发送模式 17
10.10.2 主接收模式 18
10.10.3 从接收模式 18
10.10.4 从发送模式 22
10.10.5 其它状态 26
10.10.6 I2STAT=0xF8 27

 ©2008 Guangzhou ZLGMCU Development CO., LTD.

10.10.7 I2STAT=0x00 27
10.10.8 某些特殊情况 27
10.10.9 两个主机同时启动重复起始条件 27
10.10.10 仲裁丢失后的数据传输 27
10.10.11 强制访问I²C总线 28
10.10.12 SCL或SDA低电平妨碍I²C总线的操作 28
10.10.13 总线错误 28
10.10.14 I²C状态服务程序 29
10.10.15 初始化 29
10.10.16 I²C中断服务 29
10.10.17 状态服务程序 29
10.10.18 配合实际应用的状态服务 29
10.11 软件示例 30
10.11.1 初始化程序 30
10.11.2 启动主机发送功能 30
10.11.3 启动主机接收功能 30
10.11.4 I²C中断程序 30
10.11.5 无指定模式的状态 30
10.11.6 主发送状态 31
10.11.7 主接收状态 32
10.11.8 从接收状态 33
10.11.9 从发送状态 35

 ©2008 Guangzhou ZLGMCU Development CO., LTD.

第10章 I²C总线接口

10.1 本章导读

所有 LPC1110 系列 ARM 的 I²C 模块都是相同的。

10.2 特性

- 标准 I²C 兼容总线接口，可配置为主机、从机或主/从机；
- 在同时发送的主机之间进行仲裁，而不会破坏总线上的串行数据；
- 可编程时钟允许调整 I²C 传输速率；
- 主机和从机之间的数据传输是双向的；
- 串行时钟同步允许具有不同位速率的设备通过一条串行总线进行通信；
- 串行时钟同步用作握手机制以挂起及恢复串行传输；
- 支持快速模式 Plus；
- 可识别多达 4 个不同的从机地址；
- 监控模式可观察所有的 I²C 总线通信量，而不用考虑从机地址；
- I²C 总线可用于测试和诊断；
- I²C 总线包含一个带有 2 个管脚的 I²C 兼容总线接口。

10.3 应用

与外部 I²C 标准器件相连接，如串行 RAM、LCD、音频发生器和其它微控制器等。

10.4 描述

典型的I²C总线配置如图 10.1所示。根据方向位的状态（R/W），I²C总线上可能存在以下两种类型的数据传输方式：

- 由主发送器向从接收器传输数据。主机发送的第一个字节是从机地址。接下来是数据字节数。从机每接收一个字节后返回一个应答位；
- 由从发送器向主接收器传输数据。由主机发送第一个字节（从机地址）。然后从机返回一个应答位。接下来是由从机发送数据字节到主机。主机接收到所有字节（最后一个字节除外）后返回一个应答位。接收到最后一个字节后，主机返回“非应答”位。主机设备产生所有的串行时钟脉冲及起始和停止条件。以停止或重复起始条件结束传输。由于重复起始条件也是下一次串行传输的开始，因此不释放 I²C 总线。

I²C 接口是字节导向型，有 4 个操作模式：主发送模式、主接收模式、从发送模式及从接收模式。

I²C 接口遵循整个 I²C 规范，支持在不影响同一 I²C 总线上其它器件的情况下关闭 ARM Cortex-M0。

©2008 Guangzhou ZLGMCU Development CO., LTD.

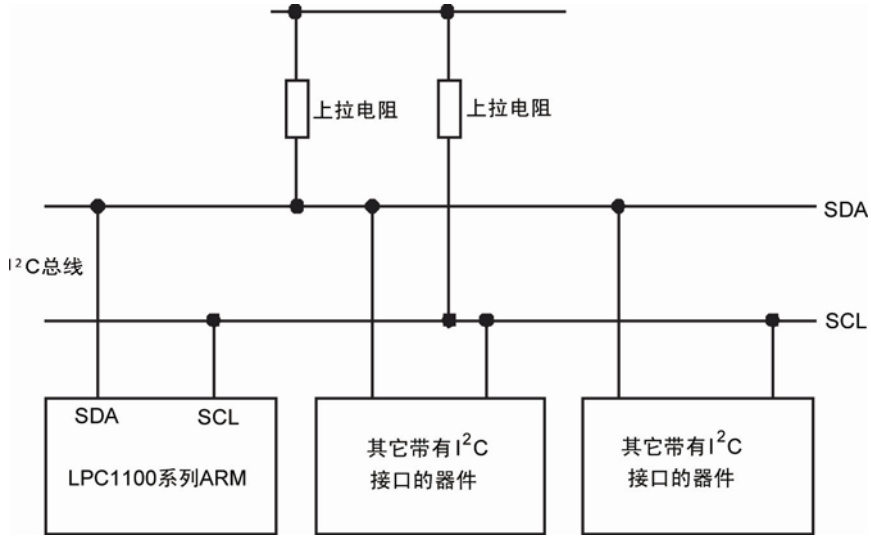


图 10.1 I²C 总线配置

10.4.1 I²C快速模式Plus

快速模式 Plus 支持 1Mbit/sec 的传输速率与 NXP 半导体现在所提供的 I²C 产品通信。

要使用快速模式 Plus，就必须正确配置 IOCONFIG 寄存器块中的 I²C 管脚，见第七章表 7.13 和表 7.14。在快速模式 Plus 中，可选择的速率在 400KHz 以上，高达 1MHz。

10.5 管脚描述

表 10.1 I²C 总线管脚描述

管脚	类型	描述
SDA	输入/输出	I²C 串行数据
SCL	输入/输出	I²C 串行时钟

I²C 总线管脚必须通过 IOCON_PIO0_4（见表 7.13）和 IOCON_PIO0_5 寄存器(见表 7.14)配置，以用于标准/快速模式或快速模式 Plus。在这些模式下，I²C 总线管脚为开漏输出并且完全兼容 I²C 总线规范。

10.6 计时和功率控制

I²C 总线接口（PCLK_I2C）的时钟由系统时钟提供（见图 3.1）。这个时钟可通过 AHBCLKCTRL 寄存器的位 5 来禁止（见第 3 章“系统 AHB 时钟控制寄存器”小节），以节省功耗。

10.7 寄存器描述

表 10.2 I²C 寄存器映射（基址 0x4000 0000）

名称	访问	地址偏移量	描述	复位值[1]
I2CONSET	R/W	0x000	**I²C 控制置位寄存器。**当向该寄存器的位写 1 时，I²C 控制寄存器中的相应位置位。写 0 时对 I²C 控制寄存器的相应位没有影响	0x00
I2STAT	RO	0x004	**I²C 状态寄存器。**在 I²C 工作期间，该寄存器提供详细的状态码，允许软件决定需要执行的下一步操作	0xF8

续上表

名称	访问	地址偏移量	描述	复位值[1]
I2DAT	R/W	0x008	**I²C 数据寄存器。**在主/从发送模式期间，要发送的数据写入该寄存器。在主/从接收模式期间，可从该寄存器读出已接收的数据	0x00
I2ADR0	R/W	0x00C	**I²C 从地址寄存器 0。**包含 7 位从地址，用于从模式下 I²C 接口操作，不用于主模式下。最低位决定从机是否对通用调用地址作出响应	0x00
I2SCLH	R/W	0x010	**SCH 占空比寄存器高半字。**决定 I²C 时钟的高电平时间	0x04
I2SCLL	R/W	0x014	**SCL 占空比寄存器低半字。**决定 I²C 时钟低电平时间。I2nSCLL 和 I2nSCLH 一起决定 I²C 主机产生的时钟频率及从模式下所用的时间	0x04
I2CONCLR	WO	0x018	**I²C 控制清零寄存器。**当向该寄存器的位写 1 时，I²C 控制寄存器中的相应位清零。写 0 对 I²C 控制寄存器中相应位没有影响	NA
I2CMMCTRL	R/W	0x01C	**监控模式控制寄存器**	0x00
I2CADR1	R/W	0x020	**I²C 从地址寄存器 1。**包含 7 位从地址，用于从模式下的 I²C 接口操作，不用于主模式下。最低位决定从机是否对通用调用地址作出响应	0x00
I2CADR2	R/W	0x024	**I²C 从地址寄存器 2。**包含 7 位从地址，用于从模式下的 I²C 接口操作，不用于主模式下。最低位决定从机是否对通用调用地址作出响应	0x00
I2CADR3	R/W	0x28	**I²C 从地址寄存器 3**。包含 7 位从地址，用于从模式下的 I²C 接口操作，不用于主模式下。最低位决定从机是否对通用调用地址作出响应	0x00
I2CDATA_BUFFER	RO	0x02C	**数据缓冲寄存器。**每次从总线接收到 9 个位（8 位数据和 ACK 或 NACK）后，I2DAT 移位寄存器的高 8 位的内容将自动传输到 DATA_BUFFER	0x00
I2CMASK0	R/W	0x030	**I²C 从地址屏蔽寄存器 0。**该屏蔽寄存器与 I2ADR0 一起决定地址匹配。当与通用调用地址（‘0000000’）比较时，屏蔽寄存器不起作用	0x00
I2CMASK1	R/W	0x034	**I²C 从地址屏蔽寄存器 1。**该屏蔽寄存器与 I2ADR0 一起决定地址匹配。当与通用调用地址（‘0000000’）比较时，屏蔽寄存器不起作用	0x00
I2CMASK2	R/W	0x038	**I²C 从地址屏蔽寄存器 2。**该屏蔽寄存器与 I2ADR0 一起决定地址匹配。当与通用调用地址（‘0000000’）比较时，屏蔽寄存器不起作用	0x00
I2CMASK3	R/W	0x03C	**I²C 从地址屏蔽寄存器 3。**该屏蔽寄存器与 I2ADR0 一起决定地址匹配。当与通用调用地址（‘0000000’）比较时，屏蔽寄存器不起作用	0x00

[1] 复位值只反映使用位中保存的数据，它不包括保留位的内容。

©2008 Guangzhou ZLGMCU Development CO., LTD.

10.7.1 I²C控制置位寄存器

I2CONSET 寄存器控制 I2CON 寄存器中位的设置，这些位控制 I²C 接口的操作。向该寄存器的位写 1 会使 I²C 控制寄存器中的相应位置位。写 0 没有影响。

表 10.3 I²C 控制置位寄存器（I2CONSET - 0x4000 0000）位描述

位	符号	描述	复位值
1:0	-	保留，用户软件不应向保留位写 1。从保留位读出的值未定义	NA
2	AA	声明应答标志	
3	SI	I²C 中断标志	0
4	STO	停止标志	0
5	STA	起始标志	0
6	I2EN	I²C 接口使能	0
7	-	保留，用户软件不应向保留位写 1。从保留位读出的值未定义	NA

I2EN：I²C 接口使能。当 I2EN 置位时，I²C 接口使能。可通过向 I2CONCLR 寄存器中的 I2ENC 位写 1 来清零 I2EN 位。当 I2EN 为 0 时，I²C 接口禁能。

当 I2EN 为“0”时，忽略 SDA 和 SCL 输入信号，I²C 块处于“不可寻址”的从状态，STO 位强制为“0”。

I2EN 不用于暂时释放 I²C 总线，因为当 I2EN 复位时，I²C 总线状态丢失。应使用 AA 标志代替。

STA：起始标志。当 STA=1 时，I²C 接口进入主模式并发送一个起始条件，如果已经处于主模式，则发送一个重复起始条件。

当 STA 为 1 且 I²C 接口没有处于主模式时，它将进入主模式，校验总线并在总线空闲时产生一个起始条件。如果总线忙，则等待一个停止条件（将释放总线）并在延迟半个内部时钟发生器周期后发送一个起始条件。当 I²C 接口已经处于主模式且已发送或接收了数据时，I²C 接口会发送一个重复起始条件。STA 可在任意时间置位，包括 I²C 接口处于可寻址的从模式时也可置位。

可通过向 I2CONCLR 寄存器中的 STAC 位写 1 来清零 STA。当 STA 为 0 时，不会产生起始条件或重复起始条件。

STA 和 STO 都置位时，如果接口处于主模式下，则向 I²C 总线发送一个停止条件，然后再发送一个起始条件。如果 I²C 接口处于从模式，则产生内部停止条件，但不发送到总线上。

STO：停止标志。在主模式下，该位置位会使 I²C 接口发送一个停止条件，或在从模式下从错误状态中恢复。当主模式下 STO=1 时，向 I²C 总线发送停止条件。当总线检测到停止条件时，STO 自动清零。

从模式下，置位 STO 位可从错误状态中恢复。这种情况下不向总线发送停止条件。硬件的表现就好像是接收到一个停止条件并切换到不可寻址的从接收模式。STO 标志由硬件自动清零。

SI：I²C 中断标志。当 I²C 状态改变时 SI 置位。但是，进入状态 F8 不会使 SI 置位，因为在那种情况下中断服务程序不起作用。

©2008 Guangzhou ZLGMCU Development CO., LTD.

当 SI 置位时，SCL 线上的串行时钟低电平持续时间扩展，且串行传输被中止。当 SCL 为高时，它不受 SI 标志的状态影响。SI 必须通过软件复位，通过向 I2CONCLR 寄存器的 SIC 位写入 1 来实现。

AA：应答标志位。当 AA 置 1 时，在 SCL 线的应答时钟脉冲内，出现下面的任意情况时都将返回一个应答信号（SDA 线为低电平）：

a) 接收到从地址寄存器中的地址。

b) 当 I2ADR 中的通用调用位（GC）置位时，接收到通用调用地址。

c) 当 I^2C 接口处于主接收模式时，接收到一个数据字节。

d) 当 I^2C 接口处于可寻址的从接收模式时，接收到一个数据字节。

可通过向 I2CONCLR 寄存器中的 AAC 位写 1 来清零 AA 位。当 AA 位为 0 时，SCL 线上的应答时钟脉冲内出现下列情况时将返回一个无应答信号（SDA 为高电平）：

a) 当 I^2C 处于主接收模式时，接收到一个数据字节。

b) 当 I^2C 处于可寻址的从接收模式时，接收到一个数据字节。

10.7.2 I^2C状态寄存器）

每个 I^2C 状态寄存器反映相应 I^2C 接口的情况。I^2C 状态寄存器为只读。

表 10.4 I^2C 状态寄存器（I2STAT - 0x4000 0004 位描述

位	符号	描述	复位值
2:0	-	这些位未使用且一直为 0	0
7:3	Status	这些位提供关于 I^2C 接口的实际状态信息	0x1F

3 个最低位总为 0。作为一个字节时，状态寄存器内容表示一个状态码。有 26 种可能存在的状态码。当状态码为 0xF8 时，没有相关信息可用且SI位没有置位。其它所有 25 个状态码符合定义的I^2C状态。当进入这些状态中的任一状态时，SI位将置位。关于状态码的完整列表，参见表 10.21~表 10.24。

10.7.3 I^2C数据寄存器

该寄存器包含要发送的数据或刚接收的数据。SI 位置位时，只有在该寄存器没有进行字节移位时，CPU 才可以对其进行读/写操作。只要 SI 位置位，I2DAT 中的数据就保持不变。I2DAT 中的数据总是从右向左移位：要发送的第一位是 MSB（位 7），接收到一个字节后，接收到数据的第一位放在 I2DAT 的 MSB 位。

表 10.5 I^2C 数据寄存器（I2DAT - 0x4000 0008）位描述

位	符号	描述	复位值
7:0	数据	该寄存器保存已接收或将要发送的数据值	0

10.7.4 I^2C从地址寄存器（I2ADR0 - 0x4000 000C）

这些寄存器可读/写，只有在 I^2C 接口设置为从模式时才可用。在主模式下，该寄存器无效。I2ADR 的 LSB 为通用调用位。当该位置位时，可识别通用调用地址（0x00）。

其中包含位 00x 的寄存器将被禁能且不与总线上的任意地址匹配。复位时 4 个寄存器都要清零到该禁能状态。

©2008 Guangzhou ZLGMCU Development CO., LTD.

表 10.6 I²C 从地址寄存器位描述

位	符号	描述	复位值
0	GC	通用调用使能位	0
7:1	地址	从模式的 I²C 器件地址	0x00

10.7.5 I²C SCL高电平占空比寄存器和低电平占空比寄存器

表 10.7 I²C SCL 高电平占空比寄存器（I2SCLH - 0x4000 0010）位描述

位	符号	描述	复位值
15:0	SCLH	SCL 高电平周期选择	0x0004

表 10.8 I²C SCL 低电平占空比寄存器（I2SCLL – 0x4000 0014）位描述

位	符号	描述	复位值
15:0	SCLL	SCL 低电平周期选择	0x0004

1. 选择适当的I²C数据速率和占空比

软件必须设定寄存器 I2SCLH 和 I2SCLL 的值以选择适当的数据速率和占空比。I2SCLH 定义了 SCL 高电平期间 PCLK_I2C 的周期数，I2SCLL 定义了 SCL 低电平期间 PCLK_I2C 的周期数。频率由下面公式得出（PCLK_I2C 是系统时钟的频率）：

$$I^2C_{bitfrequency} = \frac{PCLKI2C}{I2SCLH + I2SCLL} \qquad (8)$$

I2SCLL和I2SCLH的值必须确保数据速率在适当的I²C数据速率范围内。各寄存器的值必须大于或等于 4。表 10.9给出了根据PCLK_I2C频率和I2SCLL及I2SCLH值计算出来的I2C总线速率的示例。

表 10.9 用于选择 I²C 时钟值的 I2SCLL＋I2SCLH 值

I²C 模式	I²C 位频率	I²C CLK(MHz)								
		6	8	10	12	16	20	30	40	50
标准模式	100KHz	60	80	100	120	160	200	300	400	500
快速模式	400KHz	15	20	25	30	40	50	75	100	125
快速模式 Plus	1 MHz	-	8	10	12	16	20	30	40	50

I2SCLL 和 I2SCLH 的值不一定要相同。软件可通过设定这两个寄存器可以得到 SCL 的不同占空比。例如，I²C 总线规范定义在快速模式下和在快速模式 Plus 下的 SCL 低电平时间和高电平时间是不同的。

10.7.6 I²C控制清零寄存器

I2CONCLR 寄存器控制对 I2CON 寄存器中的位清零，这些位控制 I²C 接口的操作。向该寄存器写入 1 会清零寄存器中对应的位。向这个寄存器中写入 0 无效。

表 10.10 I²C 控制清零寄存器（I2CONCLR - 0x4000 0018）位描述

位	符号	描述	复位值
1:0	-	保留。用户软件不应向保留位写 1。从保留位读出的值未定义	NA
2	AAC	声明应答清零位	

©2008 Guangzhou ZLGMCU Development CO., LTD.

3	SIC	I^2C 中断清零位	0
4	-	保留。用户软件不应向保留位写 1。从保留位读出的值未定义	NA
5	STAC	START 标志清零位	0
6	I2ENC	I^2C 接口禁能位	0
7	-	保留。用户软件不应向保留位写 1。从保留位读出的值未定义	NA

AAC：声明应答标志清零位。向该位写 1 可清零 I2CONSET 寄存器中的 AA 位。写 0 无效。

SIC：I^2C 中断标志清零位。向该位写 1 可清零 I2CONSET 寄存器中的 SI 位。写 0 无效。

STAC：起始标志清零位。向该位写 1 可清零 I2CONSET 寄存器中的 STA 位。写 0 无效。

I2ENC：I^2C 接口禁能位。向该位写 1 可清零 I2CONSET 寄存器中的 I2EN 位。写 0 无效。

10.7.7 I^2C监控模式控制寄存器

该寄存器控制监控模式，监控模式允许 I^2C 模块在不需实际参与通信或干扰 I^2C 总线的情况下监控 I^2C 总线上的流量。

表 10.11 I^2C 监控模式控制寄存器（I2CMMCTRL0 - 0x4000 001C）位描述

位	符号	值	描述	复位值
0	MM_ENA	0 1	监控模式使能 监控模式禁能 I^2C 模块将进入监控模式。在该模式下，SDA 输出将被强制为高电平。这可防止 I^2C 模块向 I^2C 数据总线输出任何类型的数据（包括 ACK） 根据 ENA_SCL 位状态，也可以将输出强制为高电平，以防止模块控制 I^2C 时钟线	0
1	ENA_SCL	0 1	SCL 输出使能 当模块处于监控模式下，清零该位则 SCL 输出将被强制为高电平。如上所述，这可防止模块控制 I^2C 时钟线 该位置位时，I^2C 模块将以与正常操作中相同的方法控制时钟线。这意味着，作为从机设备，I^2C 模块可“延长”时钟线（使其为低电平），直到它有时间响应 I^2C 中断为止[1]	0
3	MATCH_ALL	0 1	选择中断寄存器匹配 该位清零时，只有在 4 个（最多）地址寄存器（如上面所描述的）中的一个出现匹配时，才会产生中断。也就是说，模块会作为普通的从机响应，直到有地址识别 当该位置 1 且 I^2C 处于监控模式时，可在任意接收的地址上产生中断。这将使器件监控总线上的所有通信量	0

[1] 当 ENA_SCL 位清零且 I^2C 不能再延迟总线时，中断响应时间就变得很重要。为了使器件在这些情况下能有更多时间对 I2C 中断作出响应，就需要使用 DATA_BUFFER 寄存器（参考“I^2C 数据缓冲寄存器”小节）来保存接收到的数据，保存时间为发送完一个 9 位字的时间。

注：如果 MM_ENA 为“0”（例如，如果模块没有处于监控模式），则 ENA_SCL 和 MATCH_ALL 位无效。

©2008 Guangzhou ZLGMCU Development CO., LTD.

1. 监控模式下的中断

当模块处于监控模式时所有中断将正常出现。这意味着检测到地址匹配时就会产生第一个中断（如果 MATCH_ALL 位置位，则接收到任意地址都会产生中断，否则只有在地址与 4 个地址寄存器中的一个匹配时才会产生中断）。

检测地址匹配后，对于从机写传输，每接收到一个字节就会产生中断，对于从机读传输，每发送完模块“认为”要发送的字节后产生中断。在第二种情况下，数据寄存器实际上包含了总线上其它从机发送的数据，这些从机实际上是被主机寻址的。

所有中断产生后，处理器可读数据寄存器以查看总线上实际发送的数据。

2. 监控模式下仲裁丢失

在监控模式下，I^2C 模块不能响应总线主机的信息请求或发布应答，而是由总线上的其它从机作出响应。这很可能会导致仲裁丢失。

软件应当意识到模块在监控模式中并且不应当对任何检测到的仲裁状态丢失做出响应。另外，模块中还可设计一个硬件以阻止一些/所有的仲裁丢失状态发生（如果这些状态会阻止产生想要的中断或产生不想要的中断）。是否需要附加硬件仍待定。

10.7.8 I^2C从地址寄存器

这些寄存器可读/写，只有在 I^2C 接口设置为从模式时才可用。在主模式下，该寄存器无效。I2ADR 的 LSB 为通用调用位。当该位置位时，可识别通用调用地址（0x00）。

其中包含位 00x 的寄存器将被禁能且不与总线上的任意地址匹配。复位时 4 个寄存器都要清零到该禁能状态。

表 10.12 I^2C 从地址寄存器（I2ADR[0,1,2,3] - 0x4000 00[0C,20,24,28]）位描述

位	符号	描述	复位值
0	GC	通用调用使能位	0
7:1	地址	从模式的 I^2C 器件地址	0x00

10.7.9 I^2C数据缓冲寄存器

在监控模式下，如果 ENA_SCL 没有置位，则 I^2C 模块就不能延长时钟（使总线延迟）。这意味着处理器读取总线接收数据内容的时间有限。如果处理器读 I2DAT 移位寄存器，则在接收数据被新数据覆写前，它通常只有一个位时间对中断作出响应。

为了使处理器有更多时间响应，将增加一个新的 8 位只读 DATA_BUFFER 寄存器。总线上每接收到 9 位（8 位数据加上 1 位 ACK 或 NACK）后，I2DAT 移位寄存器的高 8 位的内容将自动传输到 DATA_BUFFER。这意味着处理器有 9 位发送时间响应中断及在数据被覆写前读取数据。

处理器仍可直接读 I2DAT，I2DAT 无论如何是不会改变的。

尽管 DATA_BUFFER 寄存器主要是用于监控模式（ENA_SCL 位＝0），但它也可用于在任何操作模式下随时读取数据。

表 10.13 I^2C 数据缓冲寄存器（I2CDATA_BUFFER - 0x4000 002C）位描述

位	符号	描述	复位值
7:0	数据	该寄存器保存 I2DAT 移位寄存器中高 8 位的内容	0

10.7.10 I^2C屏蔽寄存器

©2008 Guangzhou ZLGMCU Development CO., LTD.

4 个屏蔽寄存器各包含 7 个有效位（7:1）。这些寄存器中的任一位置 1 都会使接收地址的相应位自动比较（当它与屏蔽寄存器关联的 I2ADDRn 寄存器比较时）。也就是说，决定地址匹配时不考虑 I2ADDRn 寄存器中被屏蔽的位。

复位时，所有屏蔽寄存器位清零。

与通用调用地址（“0000000”）比较时，屏蔽寄存器无效。

屏蔽寄存器的位（31:8）和位（0）未使用且不应写入值。读这些位总返回 0。

当产生地址匹配中断时，处理器必须读数据寄存器（I2DAT）以决定实际引起匹配的接收地址。

表 10.14 I^2C 屏蔽寄存器（I2MASK[0,1,2,3] - 0x4000 00[30,34,38,3C]）位描述

位	符号	描述	复位值
0	-	保留。用户软件不应向保留位写 1。读取该位总是返回 0	0
7:1	MASK	屏蔽位	0x00
31:8	-	保留。用户软件不应向保留位写 1。读取该位总是返回 0	0

10.8 I^2C操作模式

在给定的应用中，I^2C 块可作为主机、从机或同时作主机和从机。在从模式，I^2C 硬件查找其 4 个从地址中的任何一个地址及通用调用地址。如果检测到其中一个地址，则请求中断。如果处理器想成为总线主机，则在进入主模式前，硬件将一直等待，直到总线空闲，这样就不会中断可能存在的从机操作。如果在总线模式下丢失总线仲裁，则 I^2C 块将立即切换到从模式并在同一串行传输中检测自身的从地址。

10.8.1 主发送模式

该模式下，数据由主机发送到从机。在进入主发送模式前，必须按表 10.15所示初始化I2CONSET寄存器。I2EN必须置 1 以使能I^2C功能。如果AA位为 0，则当另一个器件为总线上的主机时，I^2C接口不会对任何地址作出应答，因此不能进入从模式。STA、STO和SI位必须为 0。通过向I2CONCLR寄存器中的SIC位写 1 来清零SI位。写从地址后应清零STA位。

表 10.15 用于配置主模式的 I2CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	0	-	-

发送的第一个字节包含接收器件的从地址（7 位）和数据方向位。在该模式下，数据方向位（R/W）应为 0，表示执行写操作。发送的第一个字节包含从地址和写操作位。一次发送 8 位数据。每发送完一个字节后，接收到一个应答位。输出起始和停止条件指示串行传输的起始和结束。

软件置位 STA 位时，I^2C 接口将进入主发送模式。一旦总线空闲，I^2C 逻辑就会发送起始条件。发送起始条件后，SI 位置位，I2STAT 寄存器中的状态代码为 0x08。该状态代码引导状态服务程序，将从地址和写操作位装入 I2DAT 寄存器，然后清零 SI 位。通过向 I2CONCLR 寄存器中的 SIC 位写入 1 清零 SI。

当已发送从地址和R/W位并接收到应答位后，SI位再次置位，此时，主模式下可能的状态为 0x18，0x20 或 0x38，如果从模式使能（将AA置 1），则可能为 0x68，0x78 或 0xB0。各状态码的相应操作见表 10.21~表 10.24。

©2008 Guangzhou ZLGMCU Development CO., LTD.

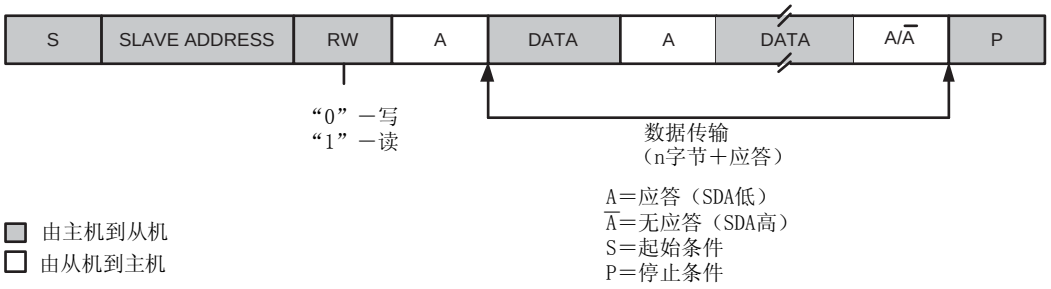


图 10.2　主发送模式下的格式

10.8.2　主接收模式

在主接收模式下，从从发送器接收数据。发起传输的方式与在主发送模式下的方式相同。发送完起始条件后，中断服务程序必须将从地址和数据方向位装入 I^2C 数据寄存器（I2DAT），然后清零 SI 位。这种情况下，数据方向位（R/W）应为 1 以指示读操作。

发送完从地址和数据方向位并接收到应答位后，SI 位置位，状态寄存器将显示状态代码。对于主模式，状态代码可能为 0x40，0x48 或 0x38。对于从模式，状态代码可能为 0x68，0x78 或 0xB0。详细信息参见

表 10.22。

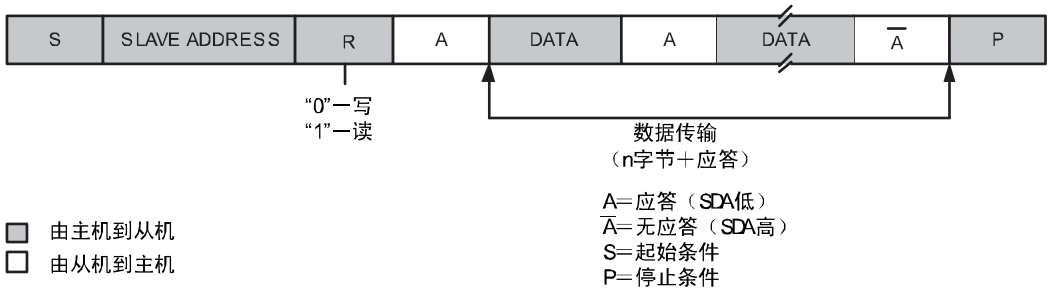


图 10.3　主接收模式下的格式

经过一个重复起始条件后，I^2C 可切换到主发送模式。

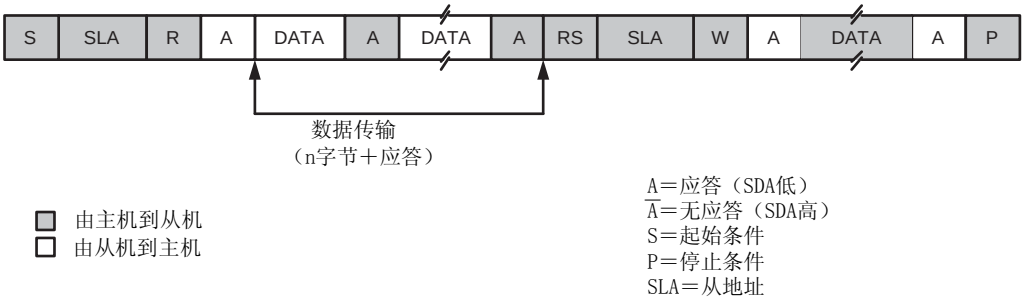


图 10.4　发送重复起始后主接收切换到主发送

10.8.3　从接收模式

从接收模式下，从主发送器接收数据字节。要初始化从接收模式，对任一从地址寄存器（I2ADR0-3）进行写操作并按表 10.16所示写I^2C控制设置寄存器（I2CONSET）。

表 10.16　用于配置从模式的 I2CONSET

位	7	6	5	4	3	2	1	0

©2008 Guangzhou ZLGMCU Development CO., LTD.

符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

I2EN 必须置 1 以使能 I^2C 功能。AA 位必须置 1 以应答其自身的从机地址或通用调用地址。STA、STO 和 SI 位置 0。

初始化 I2ADR 和 I2CONSET 后，I^2C 接口开始等待，直到被其自身地址或数据方向位后的通用地址寻址为止。如果方向位为 0（W），则进入从接收模式。如果方向位为 1（R），则进入从接收模式。接收到地址和方向位后，SI 位置位，可从状态寄存器（I2STAT）读取一个有效状态代码。关于状态代码和操作请参见

表 10.23。

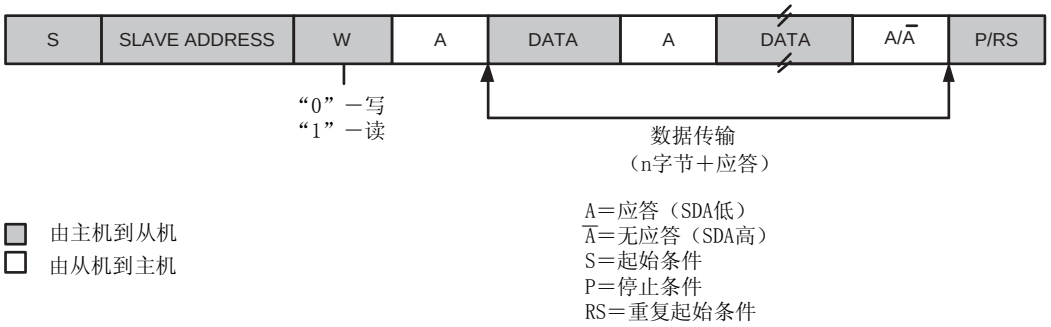


图 10.5　从接收模式的格式

10.8.4　从发送模式

接收和处理第一个字节的方式与从接收模式下相同。但是，在该模式下，方向位为 1，指示读操作。通过 SDA 发送串行数据，通过 SCL 输入串行时钟。起始和停止条件分别看作串行传输的开始和结束。在特定应用中，I^2C 可作为主机/从机。在从模式下，I^2C 硬件查寻自身从地址和通用调用地址。如果检测到其中一个地址，则请求中断。当微控制器想要变成总线主机时，在进入主模式之前，硬件开始等待，直到总线空闲为止，这样就不会中断可能存在的从机操作。如果主模式时丢失总线仲裁，则 I^2C 接口将立即切换到从模式，并可在同一串行传输中检测自身从地址。

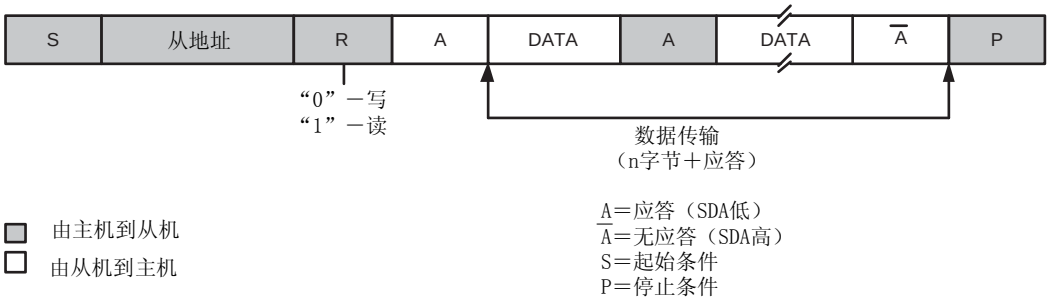


图 10.6　从发送模式的格式

10.9 功能描述

图 10.7所示为片内I^2C总线接口的执行流程，下面章节将对图中各模块进行描述。

10.9.1　输入滤波器和输出级

输入信号与内部时钟同步，小于 3 个时钟的脉冲尖峰将被滤出。

©2008 Guangzhou ZLGMCU Development CO., LTD.

I²C 输出是一个特殊端口，是为符合 I²C 规范而设计的。

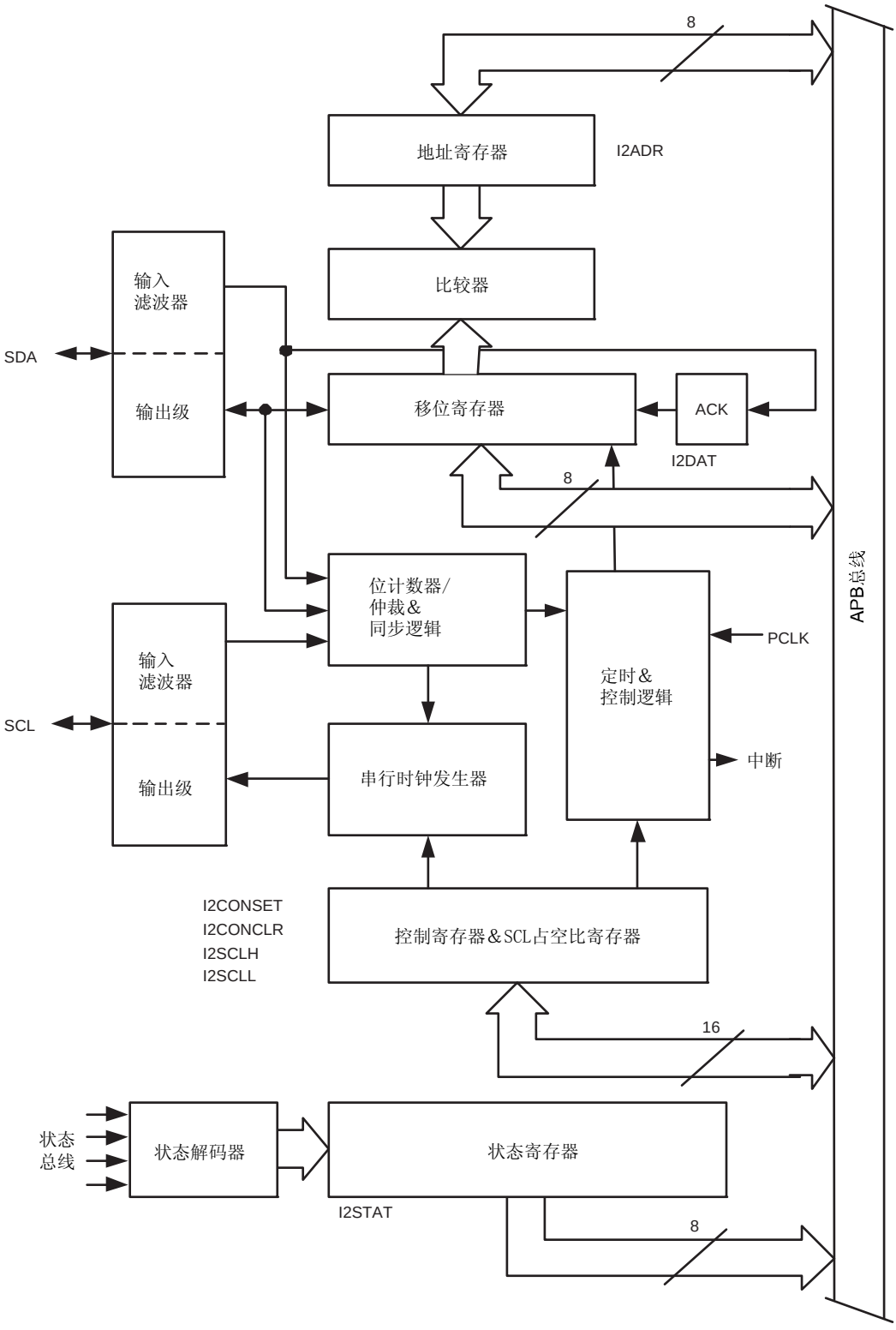


图 10.7 I²C 串行接口模块结构图

10.9.2 地址寄存器I2ADDR0~I2ADDR3

当作为从发送器或接收器时，这些寄存器可把 7 位从地址（7 个最高位）装入 I^2C 块，I^2C 块将对其作出响应。LSB（GC）用于使能通用调用地址（0x00）识别。当使能多个从地址时，接收到其自身从地址后，可从 I2DAT 寄存器读出接收到的实际地址。

10.9.3 地址屏蔽寄存器，I2MASK0~I2MASK3

4 个屏蔽寄存器各包含 7 个有效位（7:1）。当与屏蔽寄存器相关联的 I2ADDRn 寄存器相比较时，这些寄存器中的任一位置“1”都会造成接收地址中的相应位自动比较。也就是说，决定地址匹配时，不考虑 I2ADDRn 寄存器中被屏蔽的位。

如果 I2ADDRn 位 0（GC 使能位）置位且位（7:1）都为 0，则器件将对接收的地址＝“0000000“作出响应，而不考虑相关屏蔽寄存器的状态。

当产生地址匹配中断时，处理器将读数据寄存器（I2DAT）以确定实际引起匹配的接收地址。

10.9.4 比较器

比较器将接收的 7 位从地址与其自身的从地址（I2ADR 中的 7 个最高位）进行比较。它还将先接收到的 8 位字节与通用调用地址（0x00）进行比较。如果任一比较相等，则将相应状态位置位并请求中断。

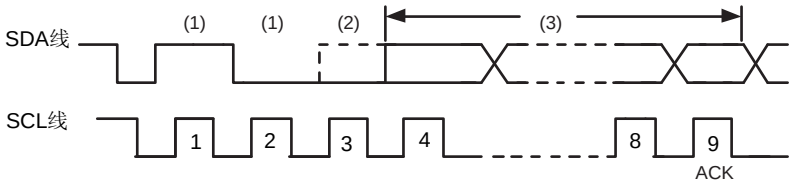
10.9.5 移位寄存器，I2DAT

这个 8 位寄存器包含一个要发送的串行数据字节或一个刚接收到的字节。I2DAT 中的数据通常是从右向左移动；要发送的第一位是 MSB（位 7），接收到一个字节后，接收到数据的第一位放置到 I2DAT 的 MSB。当数据被移出时，总线上的数据同时移入；I2DAT 通常包含总线上出现的最后一个字节。因此，在仲裁丢失时，主发送器到从接收器的转变和 I2DAT 中数据的更新同时进行。

10.9.6 仲裁及同步逻辑

在主发送模式下，仲裁逻辑校验每个发送的逻辑 1 在 I^2C 总线上是否真正以逻辑 1 出现。如果总线上另一个器件否定逻辑 1 并将 SDA 线拉低，则仲裁丢失，I^2C 块立即由主发送器转换成从接收器。I^2C 块将继续输出时钟脉冲（在 SCL 上），直到发送完当前串行字节为止。

主接收模式下也可能丢失仲裁。在该模式下，只有在I^2C块向总线返回一个“无应答：（逻辑 1）”时，才会丢失仲裁。当总线上另一个器件将该信号拉低时，仲裁丢失。由于这只会在串行字节结束时出现，因此I^2C块不再产生时钟脉冲。图 10.8所示为仲裁过程。

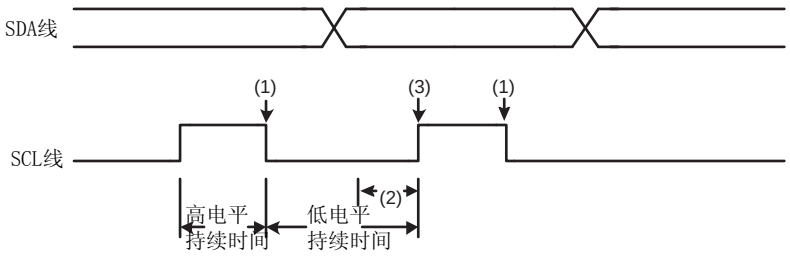


（1） 另一个器件发送串行数据。
（2） 另一个器件通过将SDA线拉低撤消了该I^2C主机发送的一个逻辑（虚线）。仲裁丢失，该 I^2C进入从接收模式。
（3） 该I^2C为从接收模式，但仍产生时钟脉冲，直到当前字节发送完为止。I^2C不会为下一个字节产生时钟脉冲。一旦赢得仲裁，SDA上的数据就由新的主机产生。

图 10.8 仲裁过程

©2008 Guangzhou ZLGMCU Development CO., LTD.

同步逻辑将使串行时钟发生器与来自另一个器件的SCL线上的时钟脉冲同步。如果有 2 个或多个主机器件产生时钟脉冲，则高电平周期由产生最短高电平持续时间的器件决定，低电平周期由产生最长低电平持续时间的器件决定。图 10.9所示为同步过程。



（1） 另一个器件在I^2C定时一个完整的高电平时间前将SCL线拉低。其它器件决定了（最短）高电平持续时间。
（2） 另一个器件在I^2C定时一个完整的低电平时间并释放SCL后继续将SCL线拉低。I^2C时钟发生器必须等待，直到SCL变为高电平。其它器件决定（最长）低电平持续时间。
（3） 释放SCL线，时钟发生器开始对高电平时间计时。

图 10.9 串行时钟同步

从机可延长低电平时间以使总线主机减速。也可以通过延长低电平时间实现握手。可在每位或一个完整字节传输后延长低电平时间。发送或接收到一个字节后，I^2C 块将延长 SCL 低电平时间且已发送应答位。设置串行中断标志（SI），继续延长低电平时间，直到串行中断清零为止。

10.9.7 串行时钟发生器

当 I^2C 块处于主发送或主接收模式时，可编程时钟脉冲发生器提供 SCL 时钟脉冲。当 I^2C 块处于从模式时，时钟脉冲发生器关闭。I^2C 输出时钟频率和占空比可通过 I^2C 时钟控制寄存器编程。详情请参见关于 I2CSCLL 和 I2CSCLH 寄存器的描述。除非总线与上面描述的其它 SCL 时钟源同步，否则输出时钟脉冲使用设定的占空比。

10.9.8 时序和控制

时序和控制逻辑为处理串行字节产生时序和控制信号。该逻辑块为 I2DAT 提供移位脉冲，可使能比较器、产生并检测起始和停止条件、接收并发送应答位，控制主/从模式，还包含中断请求逻辑并监控 I^2C 总线状态。

10.9.9 控制寄存器I2CONSET和I2CONCLR

I^2C 控制寄存器包含用于控制以下 I^2C 块功能的位：串行传输的起始和重启、串行传输终止、位速率、地址识别及应答。

I^2C 控制寄存器的内容可读出。写 I2CONSET 可置位 I^2C 控制寄存器中的相应位。反之，写 I2CONCLR 将清零 I^2C 控制寄存器中的相应位。

10.9.10 状态解码器和状态寄存器

状态解码器读取所有内部状态位并将其压缩成 5 位代码。该代码与各 I^2C 总线状态一一对应。5 位代码可用于产生向量地址，以快速处理不同的服务程序。每个服务程序处理一个特定的总线状态。如果使用 I^2C 块的所有 4 种模式，则存在 26 种可能的总线状态。当串行中断标志置位（通过硬件）并保持置位（直到中断标志被软件清零为止）时，将 5 位状态码锁存到状态寄存器的 5 个最高位。状态寄存器的 3 个最低位总为 0。如果状态码用作服务程序的向量，则程序转移到 8 位地址指向的空间。大多数的服务程序不会超过 8 字节（参见本节的软件例程）。

©2008 Guangzhou ZLGMCU Development CO., LTD.

10.10 I²C操作模式详解

I²C 接口有 4 种操作模式：

- 主发送模式；
- 主接收模式；
- 从接收模式；
- 从发送模式。

各模式下数据传输操作如图 10.10、图 10.11、图 10.12、图 10.13和图 10.14所示，表 10.17 说明了图中缩写的含义：

表 10.17 用于描述 I²C 操作的缩写

缩写	说明
S	起始条件
SLA	7 位从机地址
R	读数据位（SDA 为高电平）
W	写数据位（SDA 为低电平）
A	应答位（SDA 为低电平）
$\overline{A}$	非应答位（SDA 为高电平）
DATA	8 位数据字节
P	停止条件

图 10.10和图 10.14中，圆圈用来指示串行中断标志何时被置位。圆圈中的数字表示I2STAT寄存器中的状态代码。每当出现这些状态代码时，必须执行服务程序来继续或结束串行传输。若串行传输被挂起，这些服务程序就不再重要，直至串行中断标志被软件清除。

当进入串行中断程序时，I2STAT的状态代码用来指向跳转到的相应的服务程序。对于每个状态代码，需要的软件操作以及后面串行传输的详细情况见表 10.21~表 10.25。

10.10.1 主发送模式

在主发送模式中，向从接收器发送数据字节（见图 10.10）。在进入主发送器模式之前，I2CON必须按下表进行初始化：

表 10.18 I2CONSET 用于初始化主发送模式

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	x	-	-

I²C 速率也必须在 I2SCLL 和 I2SCLH 寄存器中配置。必须将 I2EN 设置为逻辑 1 来使能 I²C 模块。如果 AA 位复位，则当另一个器件正变成总线主机时，I²C 模块将不会应答其自身的从机地址或通用调用地址。也就是说，如果 AA 位复位，则 I²C 接口就不能进入从机模式。STA、STO 和 SI 必须复位。

此时，可通过置位 STA 位进入主发送模式。一旦总线空闲，I²C 逻辑会立即测试 I²C 总线并产生一个起始条件。当发送起始条件时，串行中断标志（SI）置位，状态寄存器（I2STAT）中的状态代码为 0x08。中断服务程序利用该状态代码进入相应的状态服务程序，将从机地址和数据方向位（SLA+W）装入 I2DAT。I2CON 的 SI 位必须在串行传输继续之前复位。

©2008 Guangzhou ZLGMCU Development CO., LTD.

当发送完从机地址和方向位且接收到一个应答位时，串行中断标志（SI）再次置位，I2STAT中可能是一系列不同的状态代码。主机模式下为 0x18，0x20 或 0x38，从机模式（AA=逻辑 1）为 0x68，0x78 或 0xB0。表 10.21中详细介绍了每个状态代码对应的操作。在发送完重复起始条件（状态 0x10）后，I^2C模块通过将SLA+R装入I2DAT切换到主接收模式。

10.10.2 主接收模式

在主接收模式中，主机所接收的数据字节来自从发送器（见图 10.11）。按主发送模式中的方法初始化传输。当发送完起始条件后，中断服务程序必须把 7 位从机地址和数据方向位（SLA+R）装入I2DAT。必须先清除I2CON中的SI位，再继续执行串行传输。

当发送完从机地址和数据方向位且接收到一个应答位时，串行中断标志 SI 再次置位，这时，I2STAT 中可能是一系列不同的状态代码。主机模式下为 0x40，0x48 或 0x38，从机模式（AA= 1）为 0x68，0x78 或 0xB0。每个状态代码对应的操作详见

表 10.22。在发送完重复起始条件（状态 0x10）后，I^2C模块通过将SLA+W装入I2DAT切换到主发送模式。

10.10.3 从接收模式

在从接收模式中，从机接收的数据字节来自主发送器（见图 10.12）。要初始化从接收模式，必须按照下表来配置I2ADR和I2CON：

表 10.19 在从接收模式中使用的 I2ADR

位	7	6	5	4	3	2	1	0
符号	自身 7 位从地址							GC

高 7 位是主机寻址时 I^2C 模块响应的地址。如果 LSB（GC）被置位，则 I^2C 模块将响应通用调用地址（0x00）；否则忽略通用调用地址。

表 10.20 初始化从接收模式时 I2CONSET 的配置

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

I^2C 总线速率的设置不影响从机模式中的 I^2C 模块。必须置位 I2EN 来使能 I^2C 模块。AA 位必须置位以使能 I^2C 模块来应答其自身从机地址或通用调用地址。STA，STO 和 SI 必须复位。

当 I2ADR 和 I2CON 完成初始化后，I^2C 模块一直等待，直至被从机地址寻址，后面跟随着数据方向位，为了工作在从接收模式中，数据方向位必须为“0”（W）。接收完其自身的从机地址和 W 位后，串行中断标志（SI）置位，可从 I2STAT 中读出一个有效的状态代码。该状态代码用作状态服务程序的向量。每个状态代码的对应操作见

表 10.23。如果I^2C模块在主机模式中仲裁丢失，也可进入从接收模式（请参考状态 0x68 和 0x78 的描述）。

如果 AA 位在传输过程中复位，则在接收完下一个数据字节后 I^2C 模块将向 SDA 返回一个非应答（逻辑 1）。当 AA 复位时，I^2C 模块不响应其自身的从机地址或通用调用地址。但是，I^2C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可临时将 I^2C 模块从 I^2C 总线上分离出来。

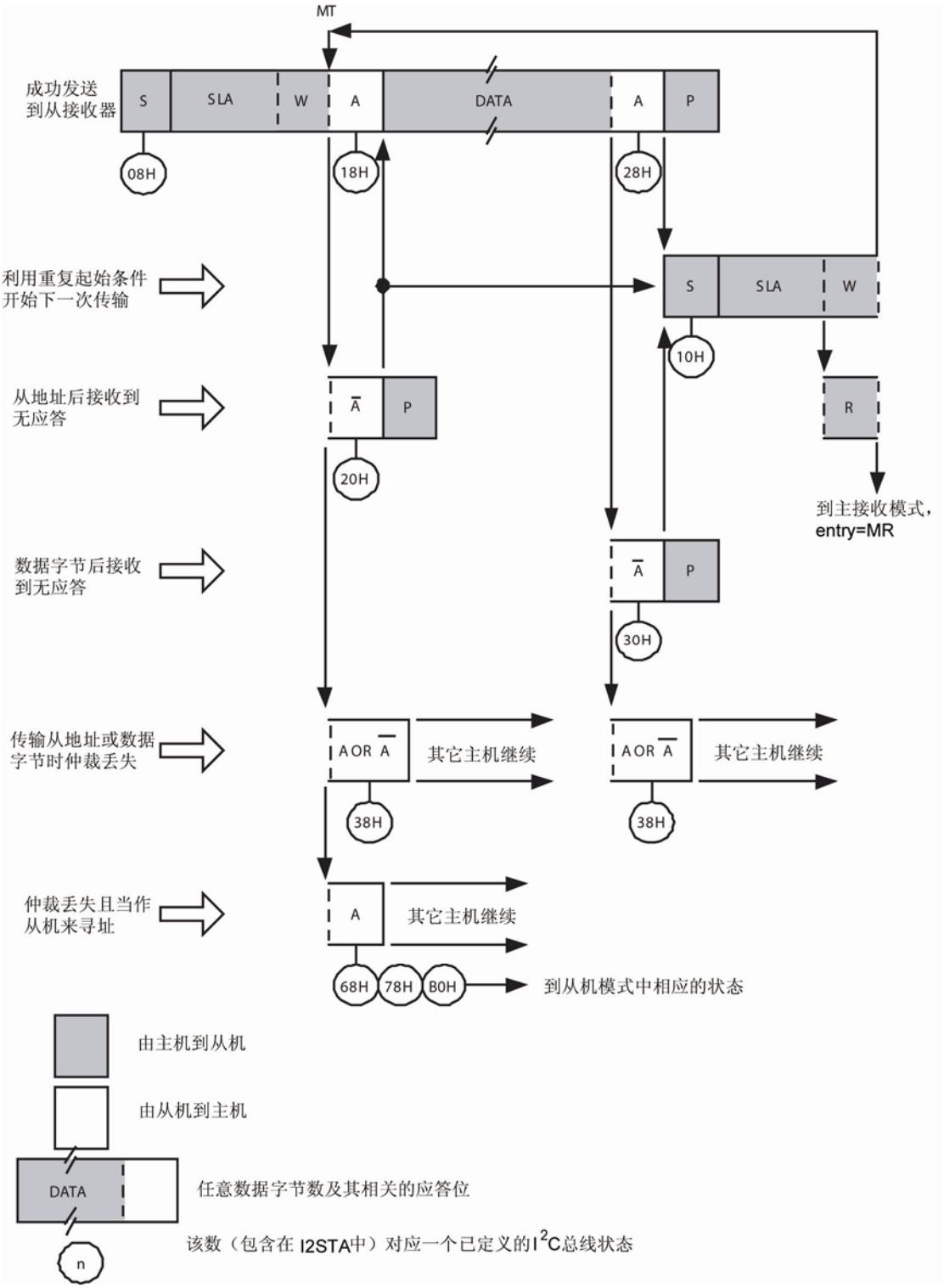


图 10.10　主发送模式下的格式和状态

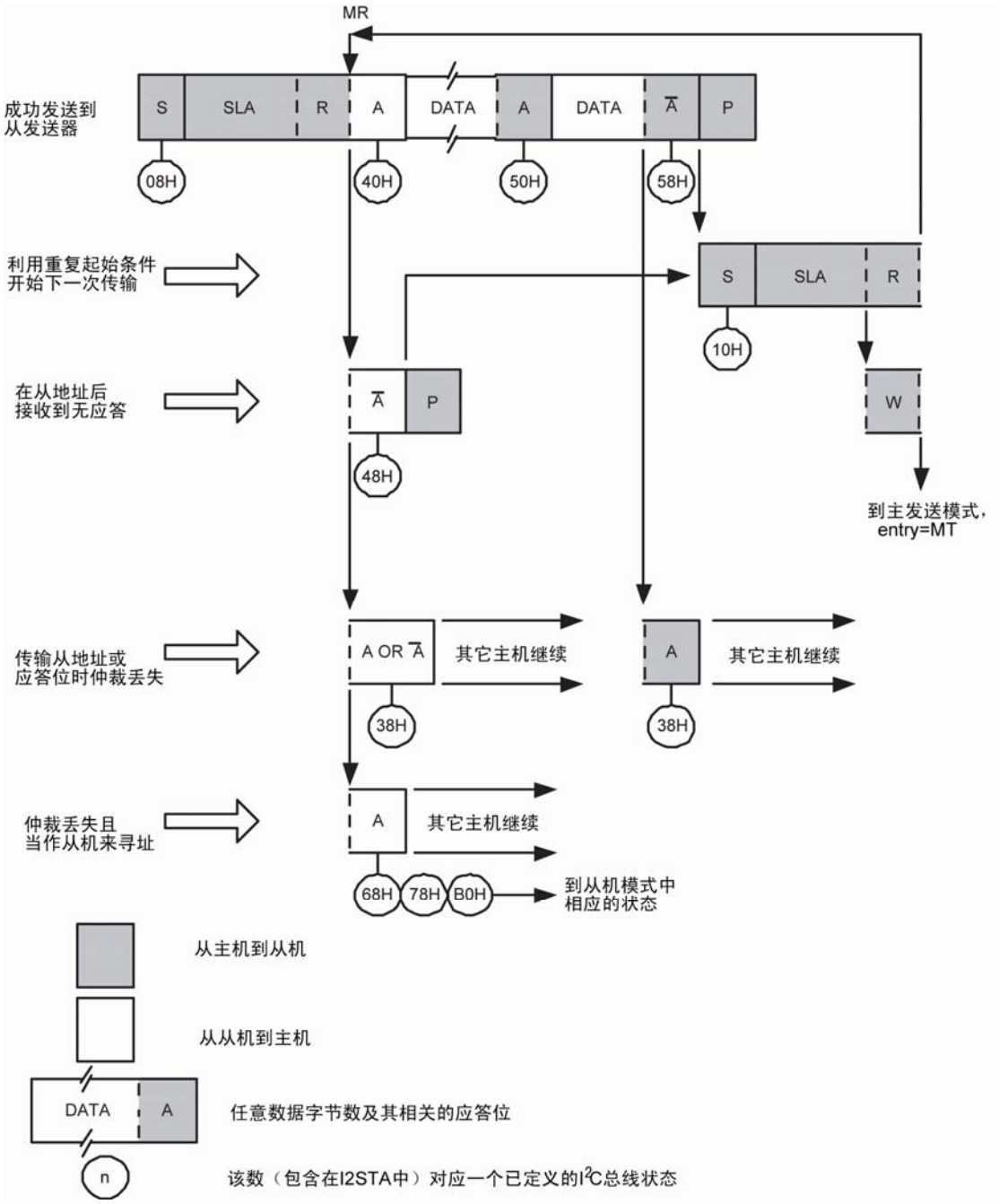


图 10.11　主接收模式下的格式和状态

©2008 Guangzhou ZLGMCU Development CO., LTD.

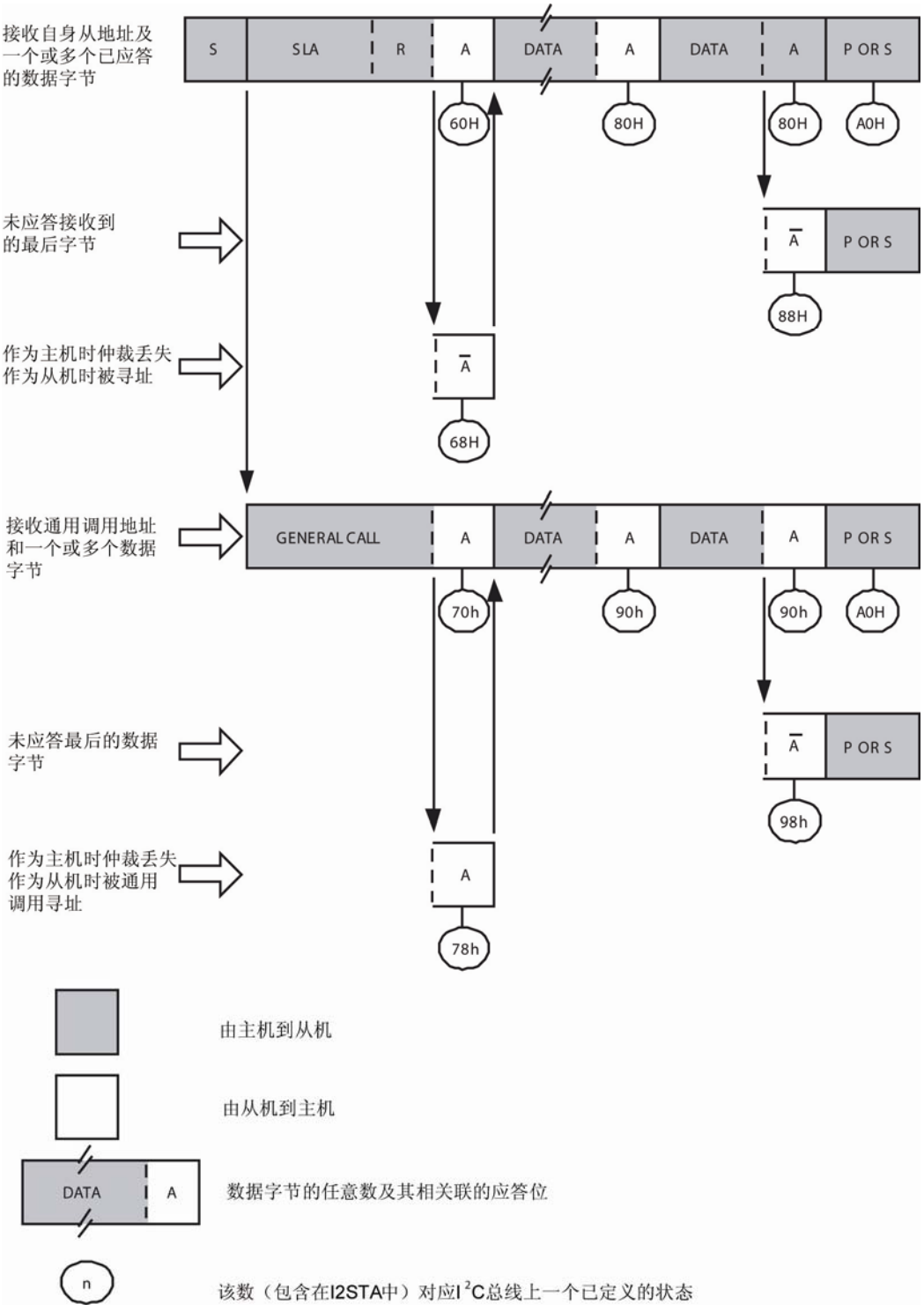


图 10.12　从接收模式下的格式和状态

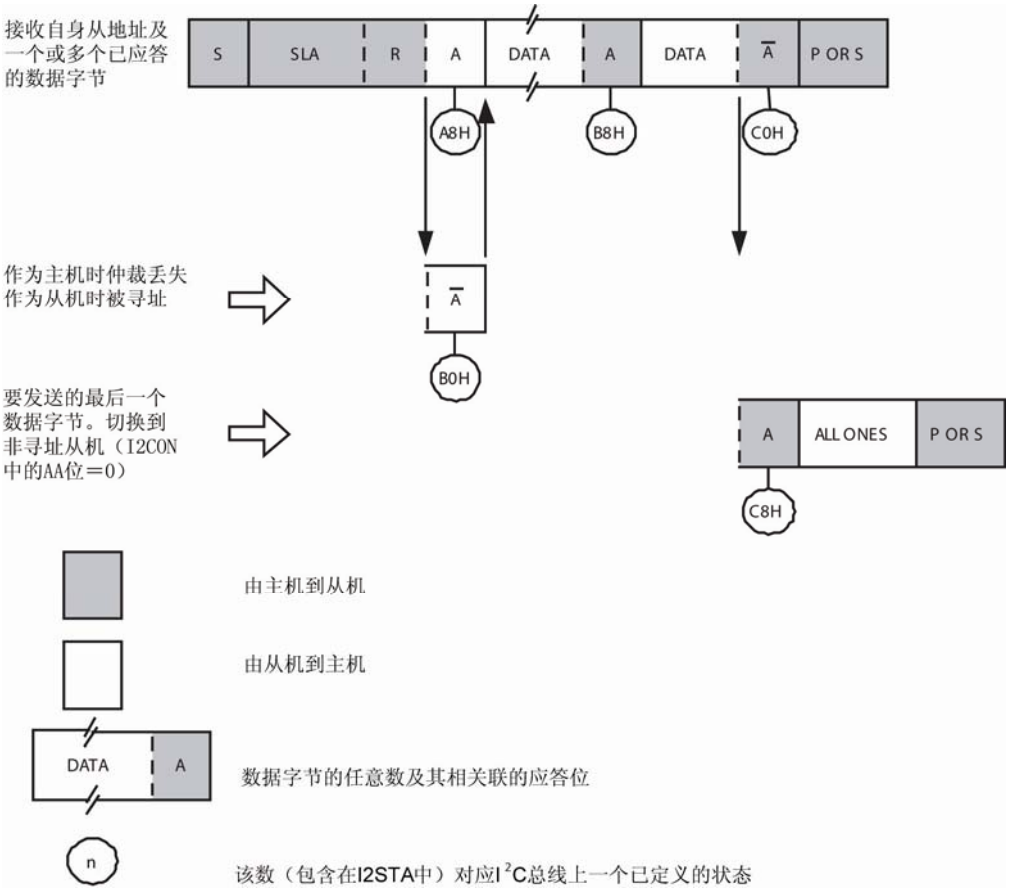


图 10.13　从发送模式下的格式和状态

10.10.4 从发送模式

在从发送模式中，向主接收器发送数据字节（见图 10.13）。数据传输按照从接收模式中的情况初始化。当初始化I2ADR和I2CON后，I^2C模块一直等待，直至被自身的从机地址寻址，之后是数据方向位，该数据方向位必须为“1”（R），以便I^2C模块工作在从发送模式下。接收完其自身的从机地址和R位后，串行中断标志（SI）置位，并且可从I2STAT中读取一个有效的状态代码。该状态代码用作状态服务程序的向量，每个状态代码的对应操作见表 10.24。如果I2C模块在主机模式下时仲裁丢失，则可进入从发送模式（见状态 0xB0）。

如果 AA 位在传输过程中复位，则 I^2C 模块将发送最后一个字节并进入状态 0xC0 或 0xC8。I^2C 模块切换到非寻址的从机模式，如果继续传输，它将忽略主接收器。因此主接收器接收所有 1 作为串行数据。当 AA 复位时，I^2C 模块不响应其自身的从机地址或通用调用地址。但是，I^2C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可用来暂时将 I^2C 模块从 I^2C 总线上分离出来。

表 10.21　主发送模式

状态代码 (I2STAT)	I^2C 总线和硬件的状态	应用软件的响应					I^2C 硬件执行的下一个操作
		写/读 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0x08	已发送起始条件	装入 SLA+W；清零 STA	X	0	0	X	将发送 SLA+W；接收 ACK 位

续上表

状态代码 (I2STAT)	I²C 总线和硬件的状态	应用软件的响应					I²C 硬件执行的下一个操作
		写/读 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0x10	已发送重复的起始条件	装入 SLA+W	X	0	0	X	同上；
		装入 SLA+R 清零 STA	X	0	0	X	将发送 SLA+W；I²C 将切换为 MST/REC 模式
0x18	已发送 SLA+W；已接收 ACK	装入数据字节	0	0	0	X	将发送数据字节，接收 ACK 位
		无 I2DAT 操作	1	0	0	X	将发送重复的起始条件
		无 I2DAT 操作	0	1	0	X	将发送停止条件；STO 标志将复位
		无 I2DAT 操作	1	1	0	X	将发送停止条件，然后发送起始条件；STO 标志将复位
0x20	已发送 SLA+W；已接收非 ACK	装入数据字节	0	0	0	X	将发送数据字节，接收 ACK 位
		无 I2DAT 操作	1	0	0	X	将发送重复的起始条件
		无 I2DAT 操作	0	1	0	X	将发送停止条件；STO 标志将复位
		无 I2DAT 操作	1	1	0	X	将发送停止条件，然后发送起始条件；STO 标志将复位
0x28	已发送 I2DAT 中的数据字节；已接收 ACK	装入数据字节	0	0	0	X	将发送数据字节，接收 ACK 位
		无 I2DAT 操作	1	0	0	X	将发送重复的起始条件
		无 I2DAT 操作	0	1	0	X	将发送停止条件；STO 标志将复位
		无 I2DAT 操作	1	1	0	X	将发送停止条件，然后发送起始条件；STO 标志将复位
0x30	已发送 I2DAT 中的数据字节；已接收非 ACK	装入数据字	0	0	0	X	将发送数据字节，接收 ACK 位
		无 I2DAT 操作	1	0	0	X	将发送重复的起始条件
		无 I2DAT 操作	0	1	0	X	将发送停止条件；STO 标志将复位
		无 I2DAT 操作	1	1	0	X	将发送停止条件，然后发送起始条件；STO 标志将复位
0x38	在 SLA+R/W 或数据字节中丢失仲裁	无 I2DAT 操作	0	0	0	X	I²C 总线将被释放；进入不可寻址从模式
		无 I2DAT 操作	1	0	0	X	当总线空闲时发送起始条件

表 10.22 主接收模式

状态代码 (I2CSTAT)	I²C 总线和硬件的状态	应用软件的响应					I²C 硬件执行的下一个操作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0x08	已发送起始条件	装入 SLA+R	X	0	0	X	将发送 SLA+R；接收 ACK 位
0x10	已发送重复的起始条件	装入 SLA+R	X	0	0	X	同上；
		装入 SLA+W	X	0	0	X	将发送 SLA+W；I²C 模块将切换为 MST/TRX 模式
0x38	在非 ACK 位中丢失仲裁	无 I2DAT 操作	0	0	0	X	I²C 总线将被释放；进入不可寻址从模式
		无 I2DAT 操作	1	0	0	X	当总线空闲时发送起始条件

续上表

状态代码 (I2CSTAT)	I²C 总线和硬件的状态	应用软件的响应					I²C 硬件执行的下一个操作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0x40	已发送 SLA+R；已接收 ACK	无 I2DAT 操作	0	0	0	0	将接收数据字节，返回非 ACK 位
		无 I2DAT 操作	0	0	0	1	将接收数据字节，返回 ACK 位
0x48	已发送 SLA+R；已接收非 ACK	无 I2DAT 操作	1	0	0	X	将发送重复的起始条件
		无 I2DAT 操作	0	1	0	X	将发送停止条件；STO 标志将复位
		无 I2DAT 操作	1	1	0	X	将发送停止条件，然后发送起始条件；STO 标志将复位
0x50	已接收数据字节，ACK 已返回	读取数据字节	0	0	0	0	将接收数据字节，返回非 ACK 位
		读取数据字节	0	0	0	1	将接收数据字节，返回 ACK 位
0x58	已接收数据字节，非 ACK 已返回	读取数据字节	1	0	0	X	将发送重复的起始条件
		读取数据字节	0	1	0	X	将发送停止条件；STO 标志将复位
		读取数据字节	1	1	0	X	将发送停止条件，然后发送起始条件；STO 标志将复位

表 10.23 从接收模式

状态代码 (I2CSTAT)	I²C 总线和硬件的状态	应用软件的响应					I²C 硬件执行的下一个操作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0x60	已接收自身的 SLA+W；已返回 ACK	无 I2DAT 操作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 I2DAT 操作	X	0	0	1	将接收数据字节，返回 ACK 位
0x68	主控器时在 SLA+R/W 中丢失仲裁；已接收自身 SLA+W，已返回 ACK	无 I2DAT 操作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 I2DAT 操作	X	0	0	1	将接收数据字节，返回 ACK 位
0x70	已接收通用调用地址（0x00）；已返回 ACK	无 I2DAT 操作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 I2DAT 操作	X	0	0	1	将接收数据字节，返回 ACK 位
0x78	主控器时在 SLA+R/W 中丢失仲裁；已接收通用调用地址，已返回 ACK	无 I2DAT 操作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 I2DAT 操作	X	0	0	1	将接收数据字节，返回 ACK 位
0x80	前一次寻址使用自身从地址；已接收数据字节；已返回 ACK	读取数据字节	X	0	0	0	将接收数据字节，返回非 ACK 位
		读取数据字节	X	0	0	1	将接收数据字节，返回 ACK 位

续上表

状态代码 (I2CSTAT)	I²C 总线和硬件的状态	应用软件的响应					I²C 硬件执行的下一个操作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0x88	前一次寻址使用自身 SLA；已接收数据字节；已返回非 ACK	读取数据字节	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		读取数据字节	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=逻辑 1，将识别通用调用地址
		读取数据字节	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		读取数据字节	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件
0x90	前一次寻址使用通用调用；已接收数据字节；已返回 ACK	读取数据字节	X	0	0	0	将接收数据字节，返回非 ACK 位
		读取数据字节	X	0	0	1	将接收数据字节，返回 ACK 位
0x98	前一次寻址使用通用调用；已接收数据字节；已返回非 ACK	读取数据字节	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		读取数据字节	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=逻辑 1，将识别通用调用地址
		读取数据字节	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		读取数据字节	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=1，将识别通用调用地址；当总线空闲后发送起始条件
0xA0	当使用 SLV/REC 或 SLV/TRX 静态寻址时，接收到停止条件或重复的起始条件	无 STDAT 操作	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 STDAT 操作	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=逻辑 1，将识别通用调用地址
		无 STDAT 操作	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 STDAT 操作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=1，将识别通用调用地址；当总线空闲后发送起始条件

©2008 Guangzhou ZLGMCU Development CO., LTD.

表 10.24 从发送模式

状态代码 (I2CSTAT)	I^2C 总线和硬件的状态	应用软件的响应					I^2C 硬件执行的下一个操作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0xA8	已接收自身的 SLA+R；已返回 ACK	装入数据字节	X	0	0	0	将发送最后一个数据字节，接收 ACK 位
		装入数据字节	X	0	0	1	将发送数据字节，接收 ACK 位
0xB0	主控器时在 SLA+R/W 中丢失仲裁；已返回 ACK	装入数据字节	X	0	0	0	将发送最后一个数据字节，接收 ACK 位
		装入数据字节	X	0	0	1	将发送数据字节，接收 ACK 位
0xB8	已发送 I2DAT 中的数据字节；已接收 ACK	装入数据字节	X	0	0	0	将发送最后一个数据字节，接收 ACK 位
		装入数据字节	X	0	0	1	将发送数据字节，接收 ACK 位
0xC0	已发送 I2DAT 中的数据字节；已接收非 ACK	无 I2DAT 操作	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 I2DAT 操作	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=逻辑 1，将识别通用调用地址
		无 I2DAT 操作	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 I2DAT 操作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=1，将识别通用调用地址；当总线空闲后发送起始条件
0xC8	I2DAT 中的最后一个数据字节已被发送（AA=0）；已接收 ACK	无 I2DAT 操作	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 I2DAT 操作	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR[0]=逻辑 1，将识别通用调用地址
		无 I2DAT 操作	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 I2DAT 操作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 I2ADR.0=1，将识别通用调用地址；当总线空闲后发送起始条件

10.10.5 其它状态

还有两种I2STAT代码与已定义的I^2C硬件状态不对应（见表 10.25），下面对这两种代码进行讨论。

©2008 Guangzhou ZLGMCU Development CO., LTD.

10.10.6 I2STAT=0xF8

这个状态码表示没有任何可用的相关信息，因为串行中断标志 SI 还没有置位。这种情况在其它状态和 I²C 模块还未开始执行串行传输之间出现。

10.10.7 I2STAT=0x00

该状态代码表示在 I²C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I²C 模块信号时也会产生总线错误。总线错误出现时 SI 置位。要从总线错误中恢复，STO 标志必须置位，SI 必须被清除。这使得 I²C 模块进入“非寻址的”从机模式（已定义的状态）并清除 STO 标志（I2CON 中的其它位不受影响）。SDA 和 SCL 线被释放（不发送停止条件）。

表 10.25 其它状态

状态代码 (I2CSTAT)	I²C 总线和硬件的状态	应用软件的响应					I²C 硬件执行的下一个操作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
0xF8	无可用的相关状态信息；SI=0	无 I2DAT 操作	无 I2CON 操作				等待或执行当前传输
0x00	由于非法起始或停止条件的出现，在 MST 或选择的从机模式中将出现总线错误。当外部干扰使 I²C 模块进入未定义的状态时也出现 0x00 状态	无 I2DAT 操作	0	1	0	X	只有 MST 或寻址的 SLV 模式中的内部硬件受影响。一般情况下，总线被释放、I²C 模块切换到非寻址的 SLV 模式。STO 复位

10.10.8 某些特殊情况

I²C 硬件可以处理串行传输过程中出现的以下几种特殊情况：

10.10.9 两个主机同时启动重复起始条件

在主发送模式或主接收模式下可以产生重复起始条件。如果此时另一个主机同时产生重复起始条件，就出现特殊情况（图 10.14）。在出现这种情况之前，任何一个主机都不会丢失仲裁，因为它们发送的数据相同。

如果 I²C 硬件在产生重复起始条件之前在 I²C 总线上检测到重复起始条件，则它将释放总线，并且不产生中断请求。如果另一个主机通过产生停止条件来释放总线，则 I²C 模块将发送一个正常的起始条件（状态 0x08），并开始重新进行完整的串行数据传输。

10.10.10 仲裁丢失后的数据传输

在主发送模式和主接收模式中仲裁可能会丢失（见图 10.8）。I2STAT寄存器中的状态代码可表示仲裁丢失，代码有：0x38，0x68，0x78 和 0xB0（见图 10.10和图 10.11）。

如果 I2CON 中的 STA 标志由服务这些状态的程序置位，则当总线再次空闲时，会发送一个起始条件（状态 0x08），并且不受 CPU 的影响，开始重新尝试完整的串行数据传输。

10.10.11 强制访问I²C总线

在某些应用中，不可控制源可能会造成总线挂起。在这种情况下，干扰、总线的暂时中断或 SDA 和 SCL 之间的暂时短路都会导致总线挂起。

如果不可控制源产生了一个多余的起始条件或屏蔽了一个停止条件，则I²C总线一直保持忙碌状态。如果STA标志置位且在相应的时间内未访问总线，那么I²C总线有可能会被强制访问。这可通过在STA标志仍被设置时置位STO标志来实现。不发送停止条件。I2C的硬件操作就好像是接收到停止条件一样，可以发送起始条件。STO标志通过硬件清零（见图 10.15）。

10.10.12 SCL或SDA低电平妨碍I²C总线的操作

如果 SDA 或 SCL 被总线上任何一个器件拉低，I²C 总线就会挂起。如果 SCL 线被总线上的器件拉低，不能继续串行传输，这时可通拉低 SCL 线的器件来处理。

一般来说，SDA线可能会被总线上另一个不和当前总线主机同步的器件拉低，不同步的原因可能是丢失了一个时钟周期或是以噪声脉冲作为时钟信号。在这种情况下，可通过向SCL发送另外的时钟脉冲来处理（见图 10.16）。虽然I²C接口没有专门用来检测总线挂起的定时器，但可以用系统的其它定时器来完成。当检测到总线挂起时，软件会强制给SCL（要求最多 9 个）时钟信号，直至SDA被器件释放。此时，从机可能还是不同步，所以还要发送一个起始条件以确保所有I²C外设同步。

10.10.13 总线错误

当格式帧的非法位置上出现起始或停止条件时总线错误产生。非法位置是指串行传输过程中的地址字节、数据位或应答位。

仅当I²C硬件作为主机或被寻址的从机进行串行传输时，它才对总线错误有反应。检测到总线错误时，I2C模块会立即切换成非寻址的从机模式，并释放SDA和SCL线，设置中断标志，并将 0x00 装入状态寄存器。该状态代码可用作状态服务程序的向量，尝试再次终止串行传输或从错误状态中恢复，如表 10.25所示。

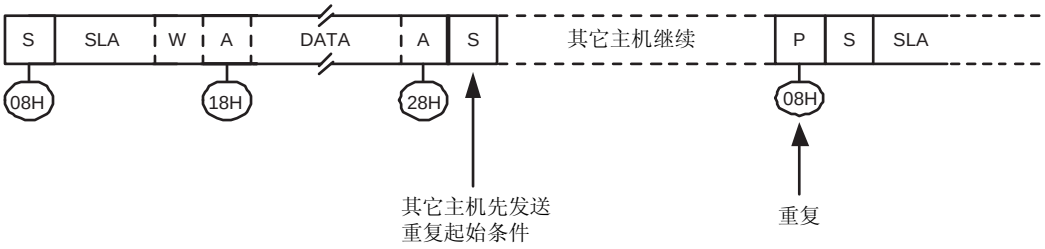


图 10.14　两个主机同时发送重复起始条件

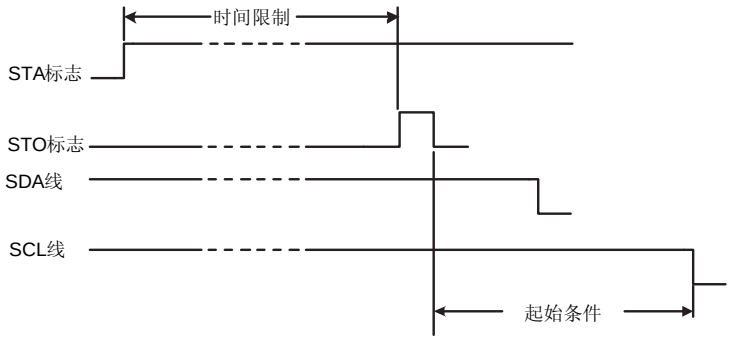
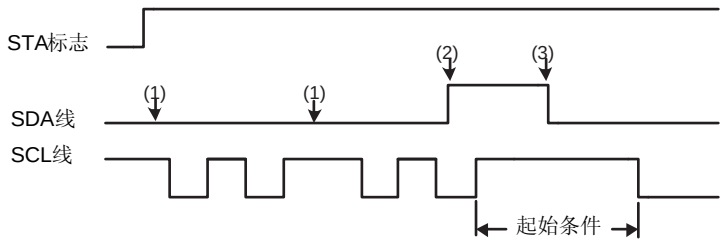


图 10.15　强制访问忙 I²C 总线

©2008 Guangzhou ZLGMCU Development CO., LTD.



（1） 尝试发送起始条件失败。
（2） 释放SDA线。
（3） 发送起始条件成功。进入08H状态。

图 10.16 从由 SDA 上低电平引起的总线干扰中恢复

10.10.14 I^2C状态服务程序

本节将介绍不同 I^2C 状态服务程序都必须执行的操作，它们包括：

[1] 复位后 I^2C 模块的初始化；
[2] I^2C 中断服务；
[3] 支持 4 种 I^2C 操作模式的 26 种状态服务程序。

10.10.15 初始化

在初始化示例中，I^2C 模块可在主机模式和从机模式中使能。对于每种模式，缓冲区可用于发送和接收数据。初始化程序将执行以下操作：

[4] 向 I2ADR 装入器件自身的从机地址和通用调用位（GC）；
[5] 置位 I^2C 中断使能位和中断优先级位；
[6] 通过同时设置 I2CON 寄存器中的 I2EN 和 AA 位来使能从机模式，通过装载 I2SCLH 和 I2SCLL 寄存器来定义串行时钟频率（主机模式）。主机程序必须从主程序开始执行。

这时，I^2C 硬件开始在 I^2C 总线上检查自身的从机地址和通用调用位。一旦检测到通用调用或自身从地址，则请求中断且把相应的状态信息装入 I2STAT。

10.10.16 I^2C中断服务

当进入 I^2C 中断时，I2STAT 含有一个状态代码，可识别要执行的 26 个状态服务中的其中一个。

10.10.17 状态服务程序

每个状态程序都是 I^2C 中断程序的组成部分，分别用来处理 26 种状态。

10.10.18 配合实际应用的状态服务

状态服务示例演示了响应 26 个 I^2C 状态代码必须执行的典型操作。如果 4 种 I^2C 操作模式中有一种或几种没被用到，则模式的相关的状态服务可被忽略，只要小心处理，那些状态就不会出现。

在应用中，可能需要在 I^2C 操作过程中执行一些超时处理，来限制无效总线或丢失服务程序。

10.11 软件示例

10.11.1 初始化程序

将 I^2C 接口初始化用作从机和/或主机的例子。

第 1 步：将自身的从机地址装入 I2ADR，使能通用调用识别（如果需要的话）。

第 2 步：使能 I^2C 中断。

第 3 步：向寄存器 I2CONSET 写入 0x44 来置位 I2EN 和 AA 位，并使能从机功能。对于主机功能，可向寄存器 I2CONSET 写入 0x40。

10.11.2 启动主机发送功能

通过建立缓冲区、指针和数据计数、然后启动起始条件来执行主发送操作。

第 1 步：初始化主机数据计数器。

第 2 步：建立数据将被发送到的从机地址，并且添加写位。

第 3 步：向 I2CONSET 写入 0x20 来置位 STA 位。

第 4 步：在主发送缓冲区内建立要发送的数据。

第 5 步：初始化主机数据计数器来匹配正在发送的信息长度。

第 6 步：退出。

10.11.3 启动主机接收功能

通过建立缓冲区、指针和数据计数、然后启动起始条件来执行主接收操作。

第 1 步：初始化主机数据计数器。

第 2 步：建立数据将被发送到的从机地址，并且添加读位。

第 3 步：向 I2CONSET 写入 0x20 来置位 STA 位。

第 4 步：建立主接收缓冲区。

第 5 步：初始化主机数据计数器来匹配接收到的信息长度。

第 6 步：退出。

10.11.4 I^2C中断程序

确定 I^2C 的状态和处理该状态的状态程序。

第 1 步：从 I2STA 中读出 I^2C 的状态。

第 2 步：使用状态值跳转到 26 个可能状态程序中的一个。

10.11.5 无指定模式的状态

1. 状态：0x00

总线错误。进入不可寻址的从机模式并释放总线。

第 1 步：向 I2CONSET 写入 0x14 来置位 STO 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

2. 主机状态

状态 08 和 10 适用于主发送模式和主接收模式。R/W 位决定了下一个状态是在主发送模式中还是在主接收模式中。

©2008 Guangzhou ZLGMCU Development CO., LTD.

3. 状态：0x08

已发送起始条件。即将发送从机地址＋R/W 位和接收 ACK 位。

第 1 步：向 I2DAT 写入从机地址和 R/W 位。

第 2 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：建立主发送模式数据缓冲区。

第 5 步：建立主接收模式数据缓冲区。

第 6 步：初始化主机数据计数器。

第 7 步：退出。

4. 状态：0x10

已发送重复起始条件。即将发送从机地址＋R/W 位和接收 ACK 位。

第 1 步：向 I2DAT 写入从机地址和 R/W 位。

第 2 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：建立主发送模式数据缓冲区。

第 5 步：建立主接收模式数据缓冲区。

第 6 步：初始化主机数据计数器。

第 7 步：退出。

10.11.6 主发送状态

1. 状态：0x18

之前状态为 8 或 10 表示已发送从机地址和写操作位，并接收了应答。即将发送第一个数据字节和接收 ACK 位。

第 1 步：将主发送缓冲区的第一个数据字节装入 I2DAT。

第 2 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：主发送缓冲区指针加 1。

第 5 步：退出。

2. 状态：0x20

已发送从机地址和写操作位并接收了非应答。即将发送停止条件。

第 1 步：向 I2CONSET 写入 0x14 来置位 STO 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

3. 状态：0x28

已发送数据并接收了 ACK。如果发送的数据是最后一个数据字节则发送一个停止条件，否则发送下一个数据字节。

第 1 步：主机数据计数器减 1，如果发送的不是最后一个数据字节就跳至第 5 步。

第 2 步：向 I2CONSET 写入 0x14 来置位 STO 和 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：退出。

第 5 步：将主发送缓冲区的下一个数据字节装入 I2DAT。

第 6 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 7 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 8 步：主机发送缓冲区指针加 1。

第 9 步：退出。

4. 状态：0x30

已发送数据并接收到非应答。即将发送停止条件。

第 1 步：向 I2CONSET 写入 0x14 来置位 STO 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

5. 状态：0x38

仲裁已在发送从机地址和写操作位或数据的过程中丢失。总线已被释放且进入非寻址的从机模式。当总线再次空闲时将发送一个新的起始条件。

第 1 步：向 I2CONSET 写入 0x24 来置位 STA 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

10.11.7 主接收状态

1. 状态：0x40

前面的状态是 08 或 10 表示已发送从机地址和读操作位，并接收到 ACK。将接收数据和返回 ACK。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

2. 状态：0x48

已发送从机地址和读操作位，并接收到非应答。将发送停止条件。

第 1 步：向 I2CONSET 写入 0x14 来置位 STO 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

3. 状态：0x50

已接收到数据，并返回 ACK。将从 I2DAT 读取数据。将接收其它的数据。如果这是最后一个数据字节，则返回非应答，否则返回 ACK。

第 1 步：读取 I2DAT 中的数据字节，存放到主机接收缓冲区。

第 2 步：主机数据计数器减 1，如果不是最后一个数据字节就跳到第 5 步。

第 3 步：向 I2CONCLR 写入 0x0C 来清除 SI 标志和 AA 位。

第 4 步：退出。

第 5 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 6 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 7 步：主机接收缓冲区指针加 1。

第 8 步：退出。

4. 状态：0x58

已接收到数据，已返回非应答。将从 I2DAT 中读取数据并发送停止条件。

第 1 步：读取 I2DAT 中的数据字节，存放到主机接收缓冲区。

第 2 步：向 I2CONSET 写入 0x14 来置位 STO 和 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：退出。

10.11.8 从接收状态

1. 状态：0x60

已接收到自身从机地址和写操作位，已返回 ACK。将接收数据和返回 ACK。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：建立从接收模式数据缓冲区。

第 4 步：初始化从机数据计数器。

第 5 步：退出。

2. 状态：0x68

用作总线主机时仲裁已在传输从机地址和 R/W 位时丢失。已接收到自身从机地址和写操作位，并已返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 STA 来重启主机模式。

第 1 步：向 I2CONSET 写入 0x24 来置位 STA 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：建立从接收模式数据缓冲区。

第 4 步：初始化从机数据计数器。

第 5 步：退出。

3. 状态：0x70

已接收到通用调用和返回 ACK。将接收数据和返回 ACK。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：建立从接收模式数据缓冲区。

第 4 步：初始化从机数据计数器。

第 5 步：退出。

4. 状态：0x78

用作总线主机时仲裁已在传输从机地址和 R/W 位时丢失。已接收到通用调用和返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 STA 来重启主机模式。

©2008 Guangzhou ZLGMCU Development CO., LTD.

第 1 步：向 I2CONSET 写入 0x24 来置位 STA 和 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：建立从接收模式数据缓冲区。

第 4 步：初始化从机数据计数器。

第 5 步：退出。

5. 状态：0x80

之前寻址自身从机地址。已接收到数据并返回 ACK。将读取其它数据。

第 1 步：读取 I2DAT 的数据字节，存放到从机接收缓冲区。

第 2 步：从机数据计数器减 1，如果不是最后一个数据字节就跳到第 5 步。

第 3 步：向 I2CONCLR 写入 0x0C 来清除 SI 标志和 AA 位。

第 4 步：退出。

第 5 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 6 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 7 步：从机接收缓冲区指针加 1。

第 8 步：退出。

6. 状态：0x88

之前寻址自身从机地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

7. 状态：0x90

之前寻址通用调用地址。已接收到数据并返回 ACK。将保存接收到的数据。只接收第一个数据字节并接收 ACK。接收其它数据字节后返回非应答。

第 1 步：读取 I2DAT 的数据字节，并放入从机接收缓冲区。

第 2 步：向 I2CONCLR 写入 0x0C 来清除 SI 标志和 AA 位。

第 3 步：退出。

8. 状态：0x98

之前寻址通用调用地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

9. 状态：0xA0

已接收停止条件或重复起始条件，但仍作为从机寻址。不保存接收到的数据。进入非寻址的从机模式。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

©2008 Guangzhou ZLGMCU Development CO., LTD.

第 3 步：退出。

10.11.9 从发送状态

1. 状态：0xA8

已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。

第 1 步：将从机发送缓冲区的第一个数据字节装入 I2DAT。

第 2 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：建立从发送模式数据缓冲区。

第 5 步：从机发送缓冲区指针加 1。

第 6 步：退出。

2. 状态：0xB0

用作总线主机时，在传输从机地址和 R/W 位时丢失仲裁。已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。当总线再次空闲后置位 STA 来重启主机模式。

第 1 步：将从机发送缓冲区的第一个数据字节装入 I2DAT。

第 2 步：向 I2CONSET 写入 0x24 来置位 STA 和 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：建立从发送模式数据缓冲区。

第 5 步：从机发送缓冲区指针加 1。

第 6 步：退出。

3. 状态：0xB8

已发送数据并接收到 ACK。将发送数据和接收 ACK 位。

第 1 步：将从机发送缓冲区的数据字节装入 I2DAT。

第 2 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 3 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 4 步：从机发送缓冲区指针加 1。

第 5 步：退出。

4. 状态：0xC0

已发送数据并接收到非应答。进入非寻址的从机模式。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

5. 状态：0xC8

已发送最后一个数据字节并接收到 ACK。进入非寻址的从机模式。

第 1 步：向 I2CONSET 写入 0x04 来置位 AA 位。

第 2 步：向 I2CONCLR 写入 0x08 来清除 SI 标志。

第 3 步：退出。

©2008 Guangzhou ZLGMCU Development CO., LTD.